TABGEN-RAG: Iterative Retrieval for Tabular Data Generation with Large Language Models

Liancheng Fang[1] **Aiwei Liu**[2] **Hengrui Zhang**[1] **Henery Peng Zou**[1] **Weizhi Zhang**[1]

Philip S. Yu[1]

[1]**University of Illinois at Chicago** [2]**Tsinghua University**
{lfang87, hzhan55, pzou3, wzhan42, psyu}@uic.edu
{liuaw20}@mails.tsinghua.edu.cn

Abstract

Large Language models (LLMs) have achieved encouraging results on tabular data generation. However, existing approaches require fine-tuning, which is computationally expensive. This paper explores an alternative: prompting a fixed LLM with in-context examples. Two main challenges arise: 1) presenting the entire training table to LLMs with limited input token length, and 2) ensuring LLMs learn effectively from the in-context examples. To address these challenges, we propose a novel retrieval-augmented generation (RAG) framework: TABGEN-RAG, to enhance the in-context learning ability of LLMs for tabular data generation. TABGEN-RAG operates iteratively, retrieving a subset of real samples that represent the residual between currently generated samples and true data. Extensive experiments on five real-world tabular datasets demonstrate that TABGEN-RAG significantly improves the quality of generated samples.

1 Introduction

Synthetic tabular data generation has been extensively studied using deep generative models, including GANs [1], VAEs [2], autoregressive models [3], and diffusion models [4–7]. These approaches face two main limitations. First, they require model retraining for each new dataset, preventing zero-shot application to unseen data. Second, they need specialized architectures to handle heterogeneous data types (continuous and categorical columns) present in tabular data. Large language models (LLMs) [8] offer a promising alternative that addresses both challenges. Their zero-shot capabilities enable data generation without domain-specific training, as they can interpret column semantics and generalize across datasets. Additionally, LLMs process both continuous and categorical data as natural language, providing a unified framework for handling heterogeneous data types. However, a critical research question remains: can LLMs generate realistic tabular data that preserves the statistical properties and relationships present in real-world datasets without fine-tuning?

[9] propose to use in-context learning to prompt LLMs to generate tabular data that follows the same distribution with real samples. At each prompting iteration, they uniformly sample a subset from the real samples and use them as in-context examples. However, we observe that LLMs can easily ignore the in-context examples and generate memorized samples from their pre-training corpora, see Figure 1(a),(b),(d). By contrast, if we provide in-context examples that have simpler patterns (e.g. have the same attribute value on a selected column), LLMs do generate faithful samples, mimicking the patterns of the in-context examples, see Figure 1(c).

Table Representation Learning Workshop at NeurIPS 2024

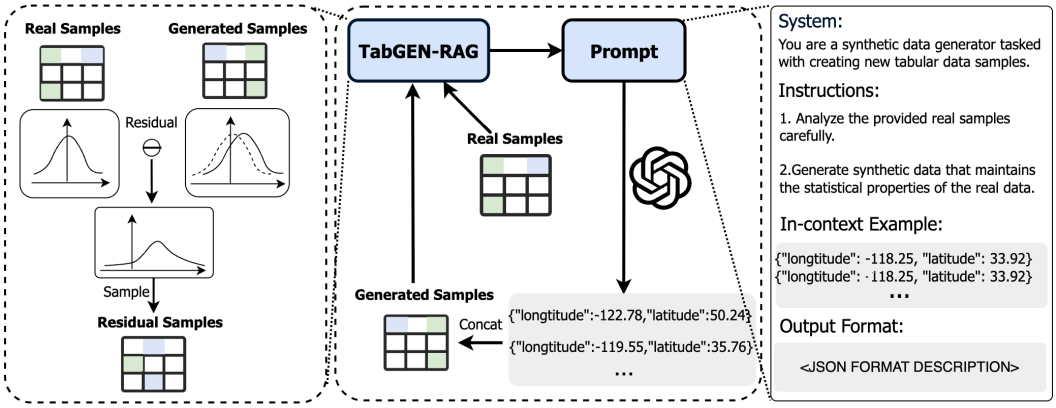
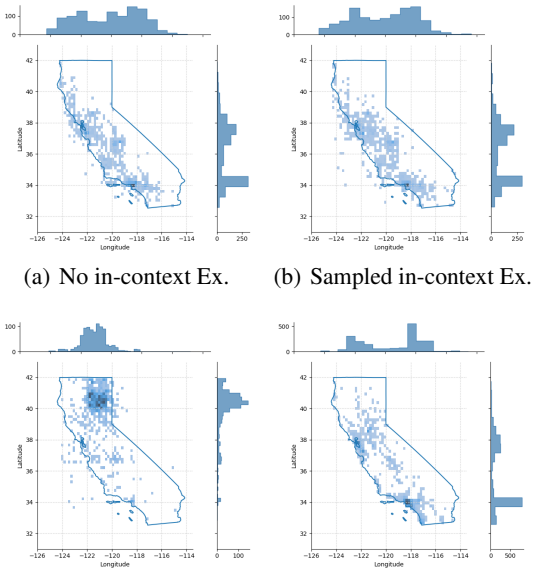


Figure 2: Overview of TABGEN-RAG framework. We generate synthetic samples in batches, at each prompt iteration, TABGEN-RAG retrieves a subset of real samples that acts as a *residual* between the currently generated samples and real data. The residual samples will be used as in-context examples to prompt LLMs in the next iteration. The full prompt template is avaliable in Appendix A.2.

This phenomenon reveals the importance of choosing in-context examples. In this work, we propose TABGEN-RAG, a novel retrieval-augmented generation (RAG) [10] framework for tabular data generation. At each prompting iteration, we retrieve a subset of real samples that has simple patterns and closely matches the gap between the current generated data distribution and the real data distribution. This approach enables 1) flexibility to choose simple patterns for in-context examples, thus ensuring that LLMs can learn from the in-context examples and generate realistic tabular data, and 2) progressively narrows the gap between the generated samples and the real data distribution, thus leading to consistent generation.



(a) No in-context Ex. (b) Sampled in-context Ex.

(c) Fixed in-context Ex. (d) Ground Truth

Figure 1: Scatter plots of 2000 California housing locations (longitude and latitude) generated under different conditions: (a) LLM output with only table headers, (b) LLM output with real data examples, (c) LLM output with range-constrained examples, and (d) actual training data. State boundary shown as solid line on each plot.

2 Methodology

Notations. Let $\mathcal{D}_{train} = \{\boldsymbol{x}_i\}_{i=1}^N$ denote a tabular dataset comprising N rows and D columns, where each row represents a data record and each column represents an attribute. The attributes can be either discrete (e.g. categorical) or continuous (e.g. numerical). We assume $\mathcal{D}_{train}$ consists of N i.i.d. samples from an unknown distribution $\mathbb{P}(\boldsymbol{x})$, where $\boldsymbol{x}$ is a D-dimensional random vector containing both numerical and categorical variables. Our objective is to generate a synthetic dataset $\mathcal{D}_{syn} = \{\hat{\boldsymbol{x}}_i\}_{i=1}^N$ where each $\hat{\boldsymbol{x}}_i$ is sampled i.i.d. from the same underlying distribution $\mathbb{P}(\boldsymbol{x})$.

2.1 In-context Examples Selection

A key challenge in using LLMs for data generation is their tendency to generate samples that follow their prior distribution, often disregarding the statistical patterns present in the in-context examples (see Figure 1). To address this limitation, we propose a novel approach that dynamically selects in-context examples representing the *residual* between the LLM's current generations and the target

distribution. This residual-based sampling aims to correct the distributional bias of the LLM. We formally define the residual as follows:

Definition 1 (Residual). Let $\boldsymbol{X}$ be a set of N $i.i.d.$ samples from a data distribution $\mathbb{P}(\boldsymbol{x})$, and let $\boldsymbol{Y}$ be an arbitrary set of samples with the same dimension as $\boldsymbol{X}$. We define the **residual** (abbrev. RES) of $\boldsymbol{X}$ w.r.t. $\boldsymbol{Y}$ as a subset of n samples of $\boldsymbol{X}$ such that, when concatenated with $\boldsymbol{Y}$, the empirical distribution of the concatenated samples is most similar to the data distribution $\mathbb{P}(\boldsymbol{x})$:

$$\texttt{RES}(\boldsymbol{X}, \boldsymbol{Y}, n) := \underset{\boldsymbol{X}' \subseteq \boldsymbol{X}, |\boldsymbol{X}'|=n}{\arg\min} \; d(\boldsymbol{X}, \boldsymbol{Y} \cup \boldsymbol{X}') \tag{1}$$

where d can be any distance metric between two empirical distributions.

Remark 1. In our case, $\boldsymbol{X}$ is the real tabular samples, $\boldsymbol{Y}$ is the current generated samples by a LLM. Intuitively, the residual samples capture the part of the real samples that LLM has not yet grasped, thus named as *residual*. We also limit the size of the residual samples with a upper-bound n, to avoid prompting LLMs with long context. In our experiments, we set $n = 500$ and instantiate d as Jensen-Shannon Divergence (JSD) and Kolmogorov-Smirnov Distance (KSD).

The brute-force way of computing the residual is computationally infeasible for large N and n. Therefore, we propose a simple heuristic to sample the residual, which empirically works well.

2.2 Compute Residual

We propose to use a simple heuristic to shrink the search space. Specifically, we first randomly select a column, then we group the real samples $\boldsymbol{X}$ based on the value of the selected column[1]. Each group of samples is then concatenated with the generated samples $\boldsymbol{Y}$. Finally, we select the group that has the smallest distance to the real samples $\boldsymbol{X}$ as the residual. The time complexity of this heuristic search algorithm is $O(N)$.

Remark 2. Note that in the heuristic search, we further constrain the residual samples to be selected from grouped samples, i.e. the subset always has the same attribute value on the selected column. This design choice can make the pattern of the residual samples more simple and consistent, thus easier for LLMs to do in-context learning.

2.3 Tabular Data Generation with TABGEN-RAG

TABGEN-RAG can be easily integrated with LLMs to generate high-quality synthetic tabular data. See Fig. 2 for an overview of the procedure. Here are the concrete steps involved in this procedure:

1. **In-context Prompting:** We plug the residual samples computed in the previous iteration into the prompt template to prompt LLMs. For the first iteration, we randomly select n samples from the real dataset $\boldsymbol{X}$ as the initial set of in-context examples. We append the generated samples into $\boldsymbol{Y}$.
2. **Residual Computation:** We then compute the residual of $\boldsymbol{X}$ w.r.t. $\boldsymbol{Y}$: $\texttt{RES}(\boldsymbol{X}, \boldsymbol{Y}, n)$. Specifically, if current iteration is a even number, we instantiate d as JSD, otherwise, we instantiate d as KSD.
3. **Iterative Refinement:** Repeat the above steps until enough synthetic samples are generated.

3 Experiments

3.1 Experiment Setup

Datasets We select five real-world tabular datasets containing both numerical and categorical attributes: **Adult, Default, Shoppers, Magic** and **California**. The statistics of the datasets are summarized in Table 2 in Appendix.

[1]For categorical columns, we group by the categorical values. For continuous columns, we discretize it into a fixed number of bins and group by the bin index.

Baselines We compare TABGEN-RAG with a variety of baselines for tabular data generation, including 1) VAE-based method TVAE [1] 2) GAN-based method CTGAN [1] 3) Diffusion-based method TabSyn [7], TabDDPM [6], CoDi [5], STaSy [4] 4) LLM-based method with fine-tuning: GReaT [3], REaLTabFormer [11], without fine-tuning: CLLM [9][2].

Evaluation Methods We evaluate the synthetic tabular data from three distinct dimensions: ① *Fidelity* - if the synthetic data faithfully recovers the ground-truth data distribution. We evaluate fidelity by 5 metrics: 1) Marginal distribution through Kolmogorov-Sirnov Test, 2) Pair-wise column correlation (Corr.) by computing Pearson Correlation, 3) Classifier Two Sample Test (C2ST) 4) Precision and Recall, 5) Jensen-Shannon Divergence (JSD). ② *Utility* - the utility of the synthetic data when used to train downstream models, we use the Train-on-Synthetic-then-Test (TSTR) protocol to evaluate the AUC score of XGBoost model on predicting the target column of each dataset. ③ *Privacy* - if the synthetic data is not copied from the real records, we employ the Distance to Closest Record (DCR) metric. We defer the full description of the metrics to Appendix A.6.

3.2 Results

Method	Marginal↓ %	Corr↓ %	Precision↓ %	Recall↓ %	C2ST↓ %	JSD↓ 10^{-2}
VAE-based						
TVAE [1]	13.59	17.32	11.65	9.11	41.72	0.63
GAN-based						
CTGAN [1]	16.36	20.33	30.65	11.41	42.90	0.91
Diffusion-based						
STaSy [4]	12.35	9.72	11.09	2.66	55.82	1.34
CoDi [5]	21.70	24.92	9.89	6.74	57.88	1.07
TabDDPM [6]	14.04	8.16	13.37	2.27	24.21	0.85
TabSyn [7]	1.40	2.36	3.76	2.29	2.64	0.05
LLM-Finetuned						
GReaT [3]	15.53	40.48	1.49	10.06	48.28	1.06
LLM-Prompt-Only						
CLLM w. GPT-4o-mini	13.17	19.57	5.54	8.08	39.02	0.78
Ours w. GPT-4o-mini	11.39	17.07	6.63	4.67	37.63	0.80
Improvement	**13.5%**	**12.8%**	**19.7%**	**42.2%**	**3.5%**	–
CLLM w. GPT-4o	10.57	13.46	4.00	4.25	31.51	0.63
Ours w. GPT-4o	9.14	12.86	4.93	2.80	26.70	0.62
Improvement	**13.6%**	**4.5%**	–	**34.1%**	**15.3%**	**1.6%**

Table 1: **Fidelity**: Comparison of various methods on fidelity metrics. Results are averaged over all datasets. All metrics are scaled to percentages (%) or 10^{-2}, and reversed so that lower values indicate better performance.

In Table 1, we present the performance comparison of all methods on **fidelity** metrics. For each metric, we report the average result on five datasets. TABGEN-RAG consistently outperforms existing LLM-based methods on fidelity metrics, which include training-free method CLLM and fine-tuning based method GReaT. Compare to CLLM, TABGEN-RAG achieves an improvement by margin of $3.5\% - 42.2\%$ on different fidelity metrics, when both employed with GPT-4o-mini, and by $1.6\% - 34.1\%$ when employed with GPT-4o. Notably, TABGEN-RAG achieves highest improvement on Recall: 42.2% on GPT-4o-mini and 34.1% on GPT-4o. Recall measures if the synthetic data covers a broad spectrum of real data, improving Recall indicates that the synthetic data is more diverse. This large improvement on Recall is anticipated as at each prompt iteration, TABGEN-RAG computes *residual* samples to cover the under-represented regions of the real data distribution, thus enriching the synthetic data's diversity. This observation further validates the effectiveness of TABGEN-RAG's *residual*-based iterative refinement mechanism. The **Utility** and **Privacy** results are reported in Table 3 and Figure 4. The detailed experiment results on each dataset is presented in Appendix A.8.

[2]CLLM was originally employed with GPT-3.5 and GPT-4, for fair comparison to our method, we employ CLLM with stronger models: GPT-4o-mini and GPT-4o.

4 Conclusion

This paper proposes TABGEN-RAG, a novel retrieval-augmented generation (RAG) framework for tabular data generation. TABGEN-RAG validates the importance to accomendate the strong prior distribution of LLMs in generating realistic and diverse synthetic data. Extensive experiments on five real-world tabular datasets and various metrics validate the effectiveness of TABGEN-RAG. We hope this work can pave the way for more research on using LLMs to generate realistic and diverse synthetic data.

References

[1] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. In *Advances in Neural Information Processing Systems*, page 7335–7345, 2019.

[2] Tennison Liu, Zhaozhi Qian, Jeroen Berrevoets, and Mihaela van der Schaar. Goggle: Generative modelling for tabular data by learning relational structure. In *International Conference on Learning Representations*, 2023.

[3] Vadim Borisov, Kathrin Sessler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators. In *International Conference on Learning Representations*, 2023.

[4] Jayoung Kim, Chaejeong Lee, and Noseong Park. Stasy: Score-based tabular data synthesis. In *International Conference on Learning Representations*, 2023.

[5] Chaejeong Lee, Jayoung Kim, and Noseong Park. Codi: Co-evolving contrastive diffusion models for mixed-type tabular synthesis. In *International Conference on Machine Learning*, pages 18940–18956. PMLR, 2023.

[6] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *International Conference on Machine Learning*, pages 17564–17579. PMLR, 2023.

[7] Hengrui Zhang, Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space. In *International Conference on Learning Representations*, 2024.

[8] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[9] Nabeel Seedat, Nicolas Huynh, Boris van Breugel, and Mihaela van der Schaar. Curated LLM: Synergy of LLMs and data curation for tabular augmentation in low-data regimes. In *Forty-first International Conference on Machine Learning*, 2024.

[10] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

[11] Aivin V. Solatorio and Olivier Dupriez. Realtabformer: Generating realistic relational and tabular data using transformers. *arXiv preprint arXiv:2302.02041*, 2023.

[12] Samuel A. Assefa, Danial Dervovic, Mahmoud Mahfouz, Robert E. Tillman, Prashant Reddy, and Manuela Veloso. Generating synthetic data in finance: Opportunities, challenges and pitfalls. In *Proceedings of the First ACM International Conference on AI in Finance*, ICAIF '20. Association for Computing Machinery, 2021. ISBN 9781450375849.

[13] Shuhan Zheng and Nontawat Charoenphakdee. Diffusion models for missing value imputation in tabular data. *arXiv preprint arXiv:2210.17128*, 2022.

[14] Mikel Hernandez, Gorka Epelde, Ane Alberdi, Rodrigo Cilla, and Debbie Rankin. Synthetic data generation for tabular health records: A systematic review. *Neurocomputing*, 493:28–45, 2022.

[15] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[17] Shengzhe Xu, Cho-Ting Lee, Mandar Sharma, Raquib Bin Yousuf, Nikhil Muralidhar, and Naren Ramakrishnan. Are llms naturally good at synthetic tabular data generation? *arXiv preprint arXiv:2406.14541*, 2024.

[18] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16: 321–357, 2002.

[19] Frank Nielsen. On the jensen–shannon symmetrization of distances relying on abstract means. *Entropy*, 21(5):485, 2019.

[20] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

A Appendix

A.1 Related works

Deep generative models for synthetic tabular data generation Generative models for tabular data have become increasingly important and have widespread applications [12–14]. For example, CTGAN and TAVE [1] deal with mixed-type tabular data generation using the basic GAN [15] and VAE [16] framework. GOGGLE [2] incorporates Graph Attention Networks in a VAE framework such that the correlation between different data columns can be explicitly learned. Recently, inspired by the success of Diffusion models in image generation, a lot of diffusion-based methods have been proposed, such as TabDDPM [6], STaSy [4], CoDi [5], and TabSyn [7].

LLMs and synthetic data generation. The collection of high-quality training data is crucial for developing advanced deep-learning models, but it is often associated with significant costs and time investment. Researchers have recently explored using pretrained large language models (LLMs) to generate synthetic datasets as an alternative approach, which sounds promising since it does not require training another deep generative model. High-quality synthetic data generated by other LLMs now play an important role in creating the pertaining corpus of next-generation LLMs. Although LLMs have been prone to generate high-quality synthetic text data, it remains a question if data-prompted LLMs can generate a large number of synthetic data that can recover the input data distribution [17]. Curated-LLM [9] demonstrates that LLMs are helpful in augmenting tabular data in low-data regimes, while their application on large-scale input data is unclear. Another work, GReaT [3], uses the GPT-2 language model to generate synthetic tabular data. However, fine-tuning the model on every new dataset requires fine-tuning.

A.2 Prompts used for generating tabular data

This prompt template is used in our experiments to generate realistic data that follows the same distribution as the given real data.

```
You are a synthetic data generator tasked with creating new tabular
    data samples that closely mirror the distribution and
    characteristics of the original dataset.

# Instruction
1. Analyze the provided real samples carefully.
2. Generate synthetic data that maintains the statistical properties
    of the real data.
3. Ensure all attributes cover their full expected ranges, including
    less common or extreme values.
4. Maintain the relationships and correlations between different
    attributes.
5. Preserve the overall distribution of the real data while
    introducing realistic variations.

# Key points to consider
- Replicate the data types of each column (e.g., numerical,
    categorical).
- Match the range and distribution of numerical attributes.
- Maintain the frequency distribution of categorical attributes.
- Reflect any patterns or trends present in the original data.
- Introduce realistic variability to avoid exact duplication.

# Real samples
{data}

# Output format:
Please present the generated data in a JSON format, structured as a
    list of objects, where each object represents a single data point
    with all attributes.
```

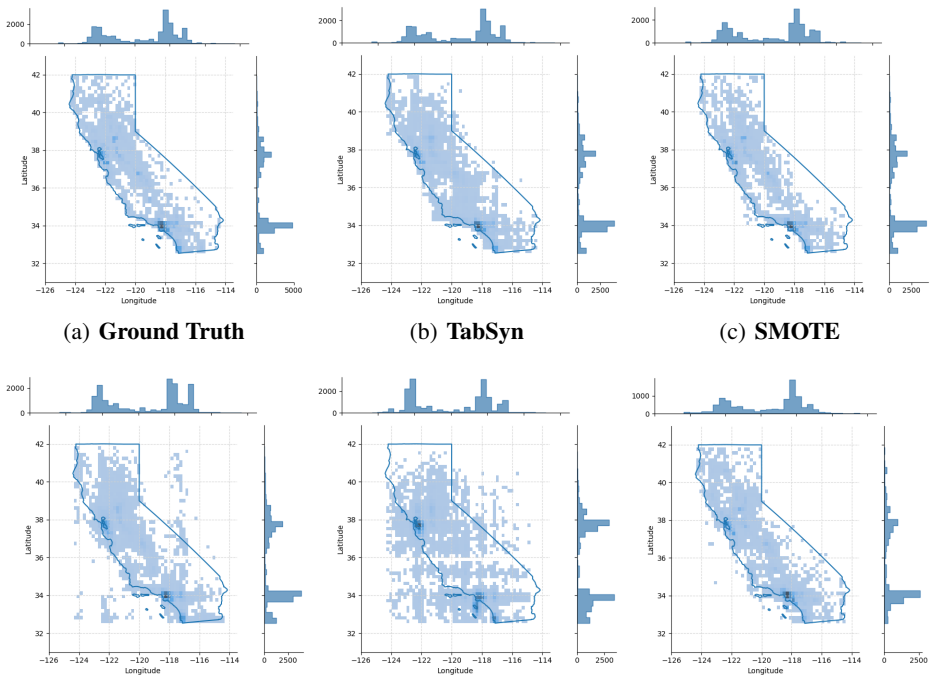
A.3 Dummy Prompt

This is a dummy prompt that we use in Figure 1 (b). No in-context example is used in this prompt.

```
You are a synthetic data generator tasked with creating new tabular
    data samples that closely mirror the distribution and
    characteristics of the original dataset.
Generate 50 samples of synthetic data.

Each sample should include the following attributes:
{attributes_list}

Make sure that the numbers make sense for each attribute.

Output Format:
Present the generated data in a JSON format, structured as a list of
    objects, where each object represents a single data point with all
     attributes.
```

A.4 Visualization

We provide a visualization of the generated samples from different methods in Figure 3. Overall, TabGen-RAG, TabSyn, and SMOTE [18] generate more realistic samples than other methods.



(a) **Ground Truth** (b) **TabSyn** (c) **SMOTE**

(d) **TVAE** (e) **CTGAN** (f) TabGen-RAG **(Ours)**

Figure 3: Joint density plots of the Longtitude and Latitude features in the California Housing dataset.

A.5 Datasets

We use five real-world datasets of varying scales, and all of them are available at Kaggle[3] or the UCI Machine Learning repository[4]. We consider five datasets containing both numberical and catergorical

[3] https://www.kaggle.com/

[4] https://archive.ics.uci.edu/

attributes: California[5], Magic[6], Adult[7], Default[8], Shoppers[9]. The statistics of these datasets are presented in Table 2.

Table 2: Statistics of datasets. # Num stands for the number of numerical columns, and # Cat stands for the number of categorical columns.

Dataset	# Rows	# Num	# Cat	# Train (In-sample)	# Test (Out-of-Sample)
California Housing	20,640	9	-	14,303	6,337
Magic Gamma Telescope	19,020	10	-	13,314	5,706
Adult Income	32,561	6	8	22,792	9,769
Default of Credit Card Clients	30,000	14	10	21,000	9,000
Online **Shoppers** Purchase	12,330	10	7	8,631	3,699

A.6 Evaluation Metrics

Fidelity To evaluate if the generated data can faithfully recover the ground-truth data distribution, we employ the following metrics: 1) **Marginal distribution:** The Marginal metric evaluates if each column's marginal distribution is faithfully recovered by the synthetic data. We use Kolmogorov-Sirnov Test for continuous data and Total Variation Distance for discrete data. 2) **Pair-wise column correlation**: This metric evaluates if the correlation between every two columns in the real data is captured by the synthetic data. We compute the Pearson Correlation between all pairs of columns then take average. In addition, we present joint density plots for the Longitude and Latitude features in the California Housing data set in Figure 3. 3) **Classifier Two Sample Test (C2ST):** This metric evaluates how difficult it is to distinguish real data from synthetic data. Specifically, we create an augmented table that has all the rows of real data and all the rows of synthetic data. Add an extra column to keep track of whether each original row is real or synthetic. Then we train a Logistic Regression classifier to distinguish real and synthetic rows. 4) **Precision and Recall:** Precision measures the quality of generated samples. High precision means the generated samples are realistic and similar to the true data distribution. Recall measures how much of the true data distribution is covered by the generated distribution. High recall means the model captures most modes/variations present in the true data. 5) **Jensen-Shannon Divergence (JSD):** This metric evaluates the Jensen-Shannon divergence [19] between the distributions of real data and synthetic data.

Utility We evaluate the utility of the generated data by accessing their performance in Machine Learning Efficiency (MLE). Following the previous works [7], we first split a real table into a real training and a real testing set. The generative models are trained on the real training set, from which a synthetic set of equivalent size is sampled. This synthetic data is then used to train a classification/regression model (XGBoost Classifier and XGBoost Regressor [20]), which will be evaluated using the real testing set. The performance of MLE is measured by the AUC score for classification tasks and RMSE for regression tasks.

Privacy A high-quality synthetic dataset should accurately reflect the underlying distribution of the original data, rather than merely replicating it. To assess this, we employ the Distance to Closest Record (DCR) metric. We begin by splitting the real data into two equal parts: a training set and a holdout set. Using the training set, we generate a synthetic dataset. We then measure the distances between each synthetic data point and its nearest neighbor in both the training and holdout sets. In theory, if both sets are drawn from the same distribution, and if the synthetic data effectively captures this distribution, we should observe an equal proportion (around 50%) of synthetic samples closer to each set. However, if the synthetic data simply copies the training set, a significantly higher percentage would be closer to the training set, well exceeding the expected 50%.

[5] https://www.kaggle.com/datasets/camnugent/california-housing-prices

[6] https://archive.ics.uci.edu/dataset/159/magic+gamma+telescope

[7] https://archive.ics.uci.edu/dataset/2/adult

[8] https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients

[9] https://archive.ics.uci.edu/dataset/468/online+shoppers+purchasing+intention+dataset

A.7 Utility and Privacy results

Method	California AUC↑	Adult AUC↑	Shoppers AUC↑	Magic AUC↑	Default AUC↑
Real	0.999	0.927	0.926	0.946	0.770
VAE-based					
TVAE [1]	0.986	0.846	0.898	0.912	0.744
GAN-based					
CTGAN [1]	0.925	0.874	0.868	0.874	0.736
Diffusion-based					
STaSy [4]	0.997	0.903	0.909	0.923	0.749
CoDi [5]	0.981	0.829	0.855	0.930	0.497
TabDDPM [6]	0.992	0.911	0.915	0.933	0.763
TabSyn [7]	0.993	0.904	0.913	0.934	0.764
LLM-Finetuned					
GReaT [3]	0.996	0.913	0.902	0.888	0.755
LLM-Prompt-Only					
CLLM w. GPT-4o-mini	0.840	0.879	0.708	0.826	0.557
TabGen-RAG w. GPT-4o-mini (Ours)	0.947	0.894	0.792	0.891	0.628
Improvement	**12.7%**	**1.7%**	**11.9%**	**7.9%**	**2.7%**
CLLM w. GPT-4o	0.947	0.891	0.865	0.885	0.718
TabGen-RAG w. GPT-4o (Ours)	0.975	0.892	0.879	0.903	0.713
Improvement	**3.0%**	**0.1%**	**1.4%**	**1.8%**	**0.5%**

Table 3: **Utility**: AUC scores of Train-on-synthetic-Test-on-real (TSTR) XGBoost model predicting the target column of each table. ↑ indicates the higher the better. – indicates that the result is missing due to training failure.

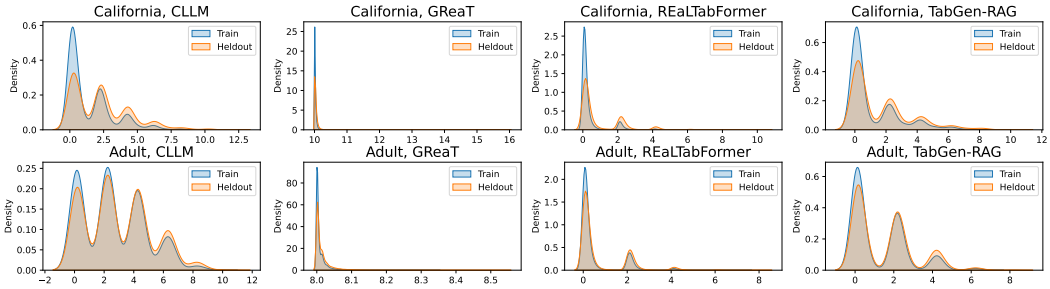


Figure 4: **Privacy:** Distributions of the DCR scores between the synthetic dataset and the training/holdout datasets. TabGen-RAG and Curated-LLM (CLLM) are both employed with GPT-4o-mini.

A.8 Detailed Fidelity results

Table 4: Error rate (%) of **column-wise density estimation**. Lower values indicate more accurate estimation (superior results).

Method	Adult↓	Default↓	Shoppers↓	Magic↓	California↓	Average↓
CTGAN	19.32± 0.77	18.25±0.82	25.71±0.12	5.69±0.17	12.84±0.29	16.36
TVAE	24.32±0.26	9.94±0.14	23.93±0.35	4.39±0.12	5.37±0.06	13.59
GReaT	12.12±0.24	19.94±0.16	14.51±0.12	16.16±0.39	10.25±0.20	14.20
STaSy	10.41±0.25	11.34±0.14	16.14±0.11	13.02±0.31	10.82±0.14	12.35
CoDi	24.84±0.36	16.54± 0.17	36.48±0.95	11.64±0.26	19.98±0.12	21.70
TabDDPM	1.32±0.13	7.59± 0.18	2.84±0.09	1.09±0.06	57.34±11.4	14.04
TabSyn	2.75±0.66	0.95±0.04	1.52±0.06	0.79±0.11	1.00±0.05	1.40
CLLM w. GPT-4o-mini	13.80±0.16	17.69±0.33	18.10±0.24	7.36±0.55	9.91±0.21	13.17
TABGEN-RAG w. GPT-4o-mini	13.19±0.16	10.99±0.33	15.10±0.24	10.12±0.55	7.54±0.21	11.54
CLLM w. GPT-4o	12.19±0.16	9.06±0.33	16.23±0.24	7.70±0.55	7.67±0.21	10.57
TABGEN-RAG w. GPT-4o	10.28±0.16	7.22±0.33	14.61±0.24	7.85±0.35	5.74±0.21	9.14

Table 5: Error rate (%) of **pair-wise column** correlation score.

Method	Adult↓	Default↓	Shoppers↓	Magic↓	California↓	Average↓
CTGAN	27.35±1.20	30.52±1.13	24.24±0.16	5.04±0.19	14.49±0.08	20.33
TVAE	36.65±4.88	19.37±0.45	20.12±0.38	4.46±0.29	5.85±2.82	17.29
GReaT	17.59±0.52	70.02±7.12	45.16±8.18	10.23±2.40	59.60±10.6	40.48
STaSy	13.50±0.25	10.65±0.26	15.29±2.15	5.48±0.23	3.59±0.51	9.72
CoDi	22.72±0.08	67.88±0.15	21.18±0.43	6.93±0.15	6.89±0.15	24.92
TabDDPM	2.50±0.25	11.55±0.27	6.23±0.16	0.57±0.12	19.83±1.09	8.16
TabSyn	4.64±0.27	3.30±0.12	2.28±0.21	0.91±0.31	0.78±0.28	2.36
CLLM w. GPT-4o-mini	25.53±0.17	26.64±0.42	22.09±0.21	10.39±0.32	13.38±0.28	19.61
TABGEN-RAG w. GPT-4o-mini	25.70±0.27	22.25±0.12	20.04±0.21	5.66±0.32	10.65±0.28	17.07
CLLM w. GPT-4o	19.55±0.27	16.65±0.12	17.89±0.21	8.16±0.32	5.18±0.28	13.49
TABGEN-RAG w. GPT-4o	17.84±0.27	18.44±0.12	14.35±0.21	6.02±0.32	7.64±0.28	12.86

Table 6: Precision score in raw scale. The higher, the better.

Method	Adult↑	Default↑	Shoppers↑	Magic↑	California↑	Average↑
CTGAN	0.8225± 0.03	0.2457± 0.04	0.7434± 0.02	0.7806± 0.07	0.8753± 0.01	0.6935
TVAE	0.9862± 0.05	0.6391± 0.04	0.8965± 0.03	0.9385± 0.02	0.9595± 0.06	0.8840
GReaT	0.9937± 0.01	0.9711± 0.02	0.9792± 0.03	0.9894± 0.01	0.9921± 0.01	0.9851
STaSy	0.9728± 0.04	0.9326± 0.05	0.9417± 0.03	0.9824± 0.02	0.6162± 0.08	0.8891
CoDi	0.7387± 0.07	0.8777± 0.02	0.9683± 0.01	0.9836± 0.01	0.9873± 0.01	0.9111
TabDDPM	0.9763± 0.02	0.7910± 0.06	0.9762± 0.02	0.9719± 0.01	0.6162± 0.04	0.8663
TabSyn	0.9739± 0.02	0.9127± 0.02	0.9764± 0.01	0.9725± 0.01	0.9763± 0.02	0.9624
CLLM w. GPT-4o-mini	0.9596± 0.05	0.9151± 0.03	0.9612± 0.02	0.9148± 0.02	0.9720± 0.02	0.9446
TABGEN-RAG w. GPT-4o-mini	0.9597± 0.02	0.8302± 0.04	0.9676± 0.01	0.9482± 0.02	0.9628± 0.03	0.9337
CLLM w. GPT-4o	0.9658± 0.01	0.9025± 0.04	0.9725± 0.02	0.9670± 0.03	0.9920± 0.01	0.9600
TABGEN-RAG w. GPT-4o	0.9569± 0.03	0.8811± 0.01	0.9692± 0.03	0.9768± 0.01	0.9697± 0.02	0.9507

Table 7: Recall score in raw scale. The higher, the better.

Method	Adult↑	Default↑	Shoppers↑	Magic↑	California↑	Average↑
CTGAN	0.9045± 0.02	0.6892± 0.04	0.8710± 0.03	0.9756± 0.01	0.9894± 0.01	0.8859
TVAE	0.7436± 0.03	0.9620± 0.02	0.9541± 0.04	0.9604± 0.01	0.9742± 0.03	0.9189
GReaT	0.8681± 0.05	0.8871± 0.01	0.8989± 0.03	0.8651± 0.02	0.9501± 0.01	0.8939
STaSy	0.9881± 0.01	0.9413± 0.04	0.9835± 0.02	0.9652± 0.01	0.9899± 0.01	0.9736
CoDi	0.9978± 0.02	0.8841± 0.03	0.8871± 0.03	0.9520± 0.02	0.9420± 0.01	0.9326
TabDDPM	0.9823± 0.01	0.9661± 0.02	0.9677± 0.03	0.9806± 0.01	0.9899± 0.01	0.9773
TabSyn	0.9808± 0.02	0.9617± 0.01	0.9783± 0.01	0.9804± 0.01	0.9841± 0.02	0.9771
CLLM w. GPT-4o-mini	0.9838± 0.03	0.7578± 0.05	0.9566± 0.01	0.9701± 0.01	0.9278± 0.02	0.9192
TABGEN-RAG w. GPT-4o-mini	0.9870± 0.02	0.8608± 0.03	0.9691± 0.02	0.9732± 0.01	0.9762± 0.01	0.9533
CLLM w. GPT-4o	0.9875± 0.02	0.9023± 0.02	0.9697± 0.01	0.9723± 0.02	0.9558± 0.01	0.9575
TABGEN-RAG w. GPT-4o	0.9922± 0.01	0.9259± 0.03	0.9753± 0.02	0.9790± 0.01	0.9874± 0.01	0.9720

Table 8: C2ST score in raw scale. The higher, the better.

Method	Adult↑	Default↑	Shoppers↑	Magic↑	California↑	Average↑
CTGAN	$0.6321_{\pm 0.04}$	$0.3540_{\pm 0.03}$	$0.5114_{\pm 0.02}$	$0.8585_{\pm 0.01}$	$0.4989_{\pm 0.04}$	0.5710
TVAE	$0.2761_{\pm 0.05}$	$0.5835_{\pm 0.02}$	$0.2996_{\pm 0.03}$	$0.8793_{\pm 0.01}$	$0.8752_{\pm 0.02}$	0.5828
GReaT	$0.9195_{\pm 0.01}$	$0.9867_{\pm 0.02}$	$0.9695_{\pm 0.01}$	$0.9992_{\pm 0.01}$	$0.9929_{\pm 0.01}$	0.9736
STaSy	$0.4598_{\pm 0.03}$	$0.5071_{\pm 0.02}$	$0.3780_{\pm 0.05}$	$0.4603_{\pm 0.04}$	$0.4539_{\pm 0.03}$	0.4518
CoDi	$0.1998_{\pm 0.02}$	$0.4763_{\pm 0.04}$	$0.1816_{\pm 0.05}$	$0.7230_{\pm 0.03}$	$0.5252_{\pm 0.01}$	0.4212
TabDDPM	$0.9605_{\pm 0.01}$	$0.8825_{\pm 0.03}$	$0.8363_{\pm 0.02}$	$0.9905_{\pm 0.01}$	$0.1199_{\pm 0.06}$	0.7579
TabSyn	$0.9195_{\pm 0.01}$	$0.9867_{\pm 0.02}$	$0.9695_{\pm 0.01}$	$0.9992_{\pm 0.01}$	$0.9929_{\pm 0.01}$	0.9736
CLLM w. GPT-4o-mini	$0.5222_{\pm 0.04}$	$0.5589_{\pm 0.03}$	$0.4085_{\pm 0.02}$	$0.7589_{\pm 0.01}$	$0.8007_{\pm 0.02}$	0.6098
TabGen-RAG w. GPT-4o-mini	$0.5493_{\pm 0.05}$	$0.5724_{\pm 0.03}$	$0.4807_{\pm 0.04}$	$0.7501_{\pm 0.01}$	$0.7661_{\pm 0.02}$	0.6237
CLLM w. GPT-4o	$0.5290_{\pm 0.03}$	$0.6780_{\pm 0.01}$	$0.5345_{\pm 0.02}$	$0.8014_{\pm 0.01}$	$0.8816_{\pm 0.02}$	0.6849
TabGen-RAG w. GPT-4o	$0.6067_{\pm 0.02}$	$0.7646_{\pm 0.03}$	$0.5752_{\pm 0.02}$	$0.8054_{\pm 0.01}$	$0.9130_{\pm 0.01}$	0.7330

Table 9: JSD in base of 10^{-2}. The lower, the better.

Method	Adult↓	Default↓	Shoppers↓	Magic↓	California↓	Average↓
CTGAN	0.38	0.114	0.066	0.056	0.182	0.091
TVAE	0.078	0.052	0.107	0.036	0.041	0.063
GReaT	0.182	0.076	0.056	0.107	0.111	0.106
STaSy	0.041	0.055	0.086	0.107	0.380	0.134
CoDi	0.073	0.067	0.103	0.142	0.152	0.107
TabDDPM	0.004	0.008	0.019	0.013	0.380	0.085
TabSyn	0.004	0.003	0.007	0.007	0.006	0.005
CLLM w. GPT-4o-mini	0.60	0.095	0.102	0.055	0.077	0.078
TabGen-RAG w. GPT-4o-mini	0.062	0.078	0.085	0.126	0.077	0.086
CLLM w. GPT-4o	0.060	0.041	0.081	0.074	0.057	0.063
TabGen-RAG w. GPT-4o	0.050	0.037	0.075	0.094	0.055	0.062